ADST/WDL/TR-93-003063

ADST
Management Command and Control Software Maintenance Manual

Loral Western Development Labs
Electronic Defense Systems Software Department
Software Engineering Laboratory
3200 Zanker road
San Jose California 95161-9041

31March 1993
Contract No.N61339-91-D-0001

CDRL A005

Simulation Training and Instrumentation Command
Naval Training Systems Center
12350 Research Parkway
Orlando, Fl 328266-3275

REPORT DOCUMENTATION PAGE

Form approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
31 March 1993

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
ADST Management Command and Control Software Maintenance Manual

5. FUNDING NUMBERS
Contract No. N61339-91-D-0001
CDRL A005

6. AUTHOR(S)
Aiken, John; Thompson, Lynn; Bright, Rick; Mitchell, Gerry; Desmeules, Peter; Kruck, Mary; Almanza, Joe

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Loral Systems Company
ADST Program Office
12443 Research Parkway, Suite 303
Orlando, FL 32826

8. PERFORMING ORGANIZATION REPORT NUMBER
ADST/WDL/TR-93-003063

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Simulation, Training and Instrumentation Command
STRICOM
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3275

10. SPONSORING ORGANIZATION REPORT

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)

This manual provides an overview ot the Management Command and Control system and its design.

14. SUBJECT TERMS

15. NUMBER OF PAGES
47

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

17. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

17. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

TABLE of CONTENTS

LIST of FIGURES

1 Introduction

1.1 Scope

This document is provided to help a software developer get an overview of the MCC System and it's software design. The document covers all parts of both MCC systems (the Masscomp MCC Host and associated Macintosh consoles and the MIPS MCC Host and associated Macintosh SCC console).

1.1.1 Software Versions

This document is valid for the following MCC software versions:

	BDS-D Version	BBN Version
MIPS MCC Host	1.0.0	4.0.4
Macintosh AIRNET SCC console	1.0.0	7.2.30d
Masscomp MCC Host	1.0.0	6.5.3
Macintosh SCC console	1.0.0	6.5.3 b
Macinstoh Admin Console	1.0.0	6.5.3 b
Macintosh Maint Console	1.0.0	6.5.3 b
Macintosh FSE Console	1.0.0	6.5.3 b
Macintosh CEC Console	1.0.0	6.5.3 b

1.2 Purpose

The purpose of this document is to describe the SIMNET Management Command and Control (MCC) software architecture as it exists.

The remainder of this document is organized as follows:

- Section 1.3, Referenced Documents, identifies all of the documents that are included in this document by reference or that provide additional information to describe the MCC software or hardware architecture.

- Section 2, Overview, includes an architectural overview of the MCC System.

- Section 3, Hardware Configuration, describes the hardware and communications interfaces involved with the MCC system.

- Section 4, Software Configuration describes the software that resides on each piece of hardware on the MCC system.

- Section 5, Notes, provides additional information that aids in understanding the MCC system described herein. It also includes a list of acronyms and definitions of terms used in this document.

1.3 Referenced documents.

The following documents are referenced in this document or provide supplementary information that will be useful in understanding the software structure of the MCC system.

The SIMNET MCC system, BBN Report No. 6473, BBN, March 31, 1987.

Reconfigurable MCC, BBN Report No. 7734, BBN, June 26, 1992.

The SIMNET Management, Command and Control System, Report No. 6473, Bolt Beranek and Newman, March 1987.

The Software Requirements & Interface Specification for the AIRNET MCC Comanche Support and Digital Message/Communications Upgrade, December 18, 1992.

The SIMNET Network and Protocols, Report No. 7627, Arthur R. Pope; Prepared for DARPA by Bolt, Barenek and Newman, Inc. June 1991.

2 Overview

2.1 Architectural Overview

Figure 2.1 portrays the MCC equipment configuration at the AIRNET Fort Rucker facility.

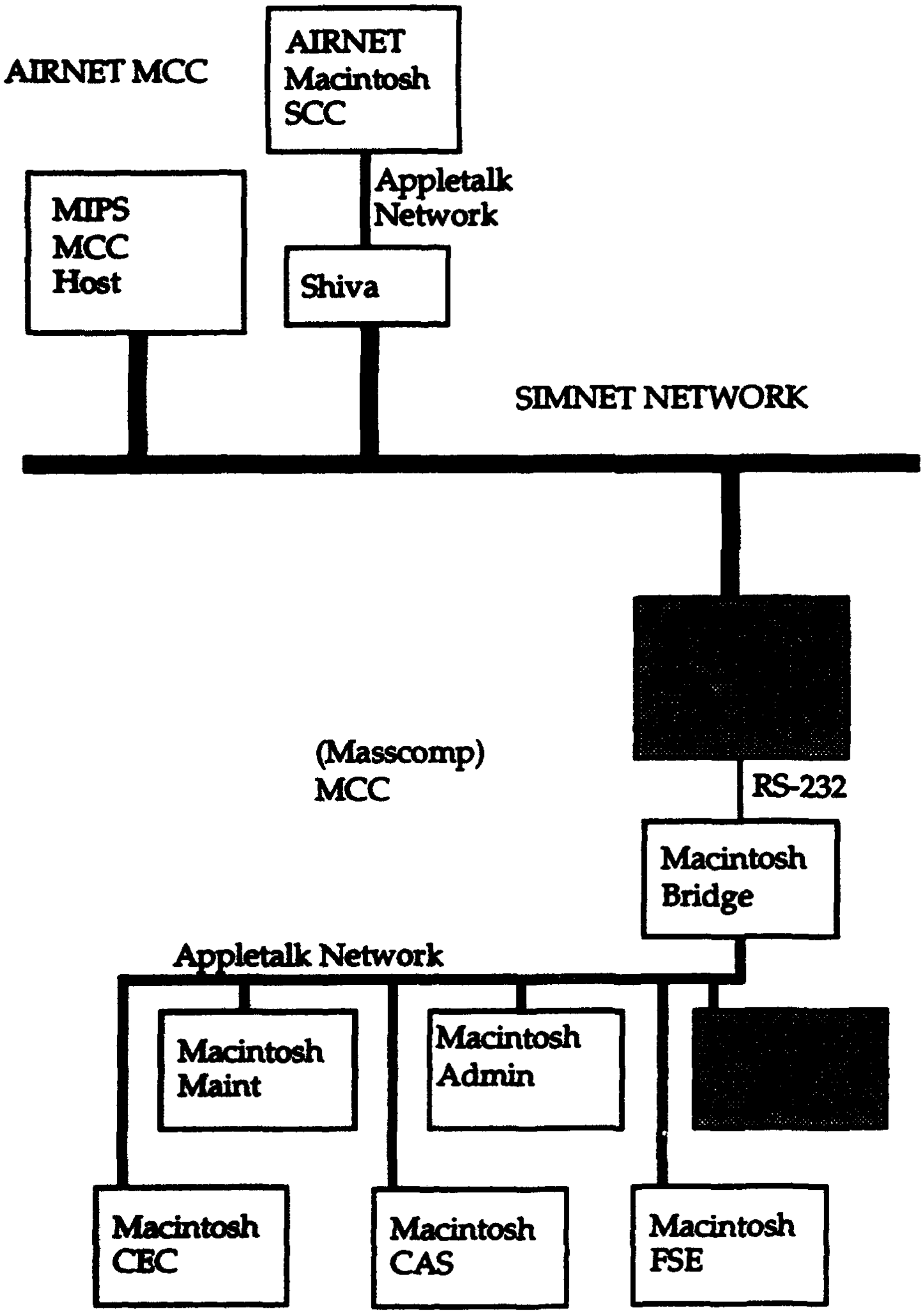


Figure 2.1 MCC System Top Level Hardware Configuration

3 Hardware Configuration

3.1 Hardware Description

The MCC is a distributed system composed of several physically separate computers. Figure 2.1 shows the hardware components and how they are interconnected.

3.1.1 CPU Configuration

3.1.1.1 MCC Console

Each of the MCC consoles is implemented by a dedicated Macintosh computer. Since most of the interfaces between the user and his console are handled locally by the dedicated Macintosh, each of these computers runs on its own software implementation. The following are standard configuration for the MCC Macintosh consoles:

- Bit-mapped screen and a mouse.
- Operating system and Toolbox software package reside in ROM to support the user interface software.
- Motorola MC68000 microprocessor capable of executing .5 MIPS.
- 512K to 1M bytes of RAM.
- A 512 x 342 pixel monochrome display.
- A floppy disc drive.
- Built in interface to support the AppleTalk network.

3.1.1.2 MCC Hosts

3.1.1.2.1 MASSCOMP MCC Host

A Masscomp 5600 system is used as the MCC host computer. The following is standard configuration for the MCC Masscomp host:

- Multibus backplane with space for several I/O interface cards.
- A version of the UNIX Operating system enhanced to support the real-time applications.
- Motorola MC68020 microprocessor capable of executing 2.5 MIPS.
- Motorola MC68881 floating point coprocessor for arithmetic calculations.
- 2 megabytes of main memory.
- An 85 megabyte disc drive.
- A floppy disc drive.
- A cartridge tape drive.
- An 8-channel RS-232 interface.
- An Ethernet Local Area Network interface.
- A TTL-level signal interface.

The 85 megabyte disc drive holds the operating system, various application programs that run on the MCC host, and copies of the terrain databases. There is enough disc space for up to three terrain databases, each covering an area of approximately 50 x 50 kilometers. The cartridge tape drive is used to import the MCC system and the new terrain databases. The floppy disc drive is used to install updates to the operating system and application programs.

The Communication Machinery Corporation (CMC) ENP-30 controller card supports the Ethernet physical and data link layer protocols. The card has a Motorola MC68000 microprocessor and 128K bytes of RAM that can be programmed to support protocols at higher layers.

The Burr-Brown Corporation MP830-71 TTL-level signal interface card provides 72 I/O lines for TTL-level signals. Sevens of these lines are used to control voice radios.

The Masscomp 5600 has a built-in RS-232 port for use in attaching a console terminal. The MCC host terminal, which may be any standard ASCII terminal, is connected to this port on the MCC host. An additional 8 RS-232 ports are supported by an High Performance Serial Multiplexor (HPSM) interface device supplied by Masscomp. None of these additional eight RS-232 ports are used in the MCC configuration.

3.1.1.2.2 MIPS MCC Host

A MIPS Magnum 3000/33 system is used as the MCC host computer. The following is standard configuration for the MCC MIPS host:

- RISC/OS Release 4.51 MIPS Version UMIPS.
- CPU: MIPS R3000 Processor chip Revision: 3.0.
- FPU: MIPS R3010 VLSI Floating Point chip Revision: 4.0.
- 64 megabytes of main memory.
- Two 200MB internal discs, one 1GB external disc.
- The commercial off-the-shelf Shiva FastPath Gateway is used.
- A single Ethernat LAN interface using TCP/IP and ENP30 protocol simultaneously.
- A cartridge tape drive.

The MIPS computer with greater computational power make it possible to use the existing SAF code to maximum effect. It provides the computational power required to make the MCC vehicles visible and vulnerable at all times.

All interaction with the simulation network is performed by the MIPS MCC. Any information that the MCC consoles need about events occuring in a simulated exercise is transmited from the MIPS MCC over a network link to the MCC consoles. Vehicle locations, status, capabilities, and the status of remote simulation entities are all maintained by the MIPS MCC; it acts as a filter and sends to the MCC consoles only the information that they actually need or request.

3.1.2 Computer Peripherals

3.1.2.1 AppleTalk Network

An AppleTalk network links the MCC consoles to the MCC host. The following are features of the AppleTalk network:

- The raw data rate is 230.4 kilobits per second over a maximum cable length of 300 meters.
- The cable is shielded, twisted pair.
- Up to 32 devices may be attached to the network.
- The role of each device in supporting the network is passive; should any device fail, communication among other devices are not affected.

There is no direct hardware connection between the Masscomp MCC host and the AppleTalk network. An intermediary Macintosh called the Bridge is used to provide communication between devices across the AppleTalk network. The MCC system uses the service of two AppleTalk protocols: the Name Binding Protocol (NBP) and the AppleTalk Transaction Protocol (ATP). The ;e protocol, in turn, use lower layer protocols called the Datagram Delivery Protocol (DDP), and the AppleTalk Link Access Protocol (ALAP). Figure 3.1.2.1 portrays the relationship between these protocols.

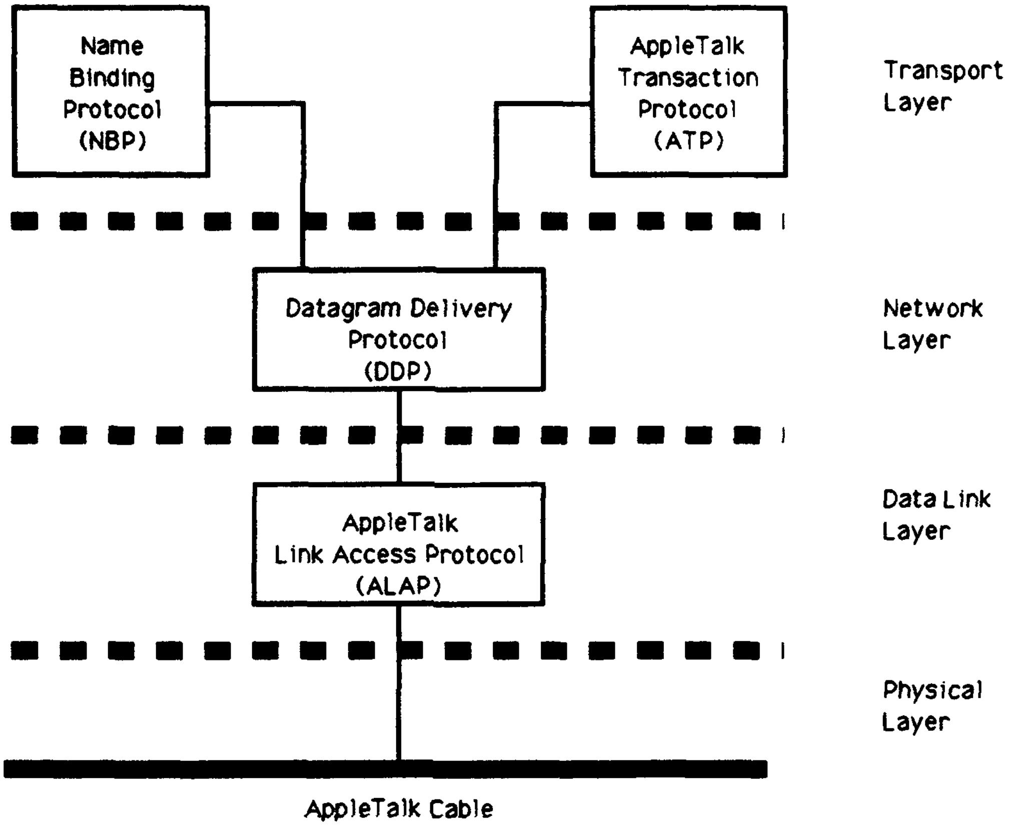

Name Binding Protocol (NBP)
AppleTalk Transaction Protocol (ATP)
Transport Layer
Datagram Delivery Protocol (DDP)
Network Layer
AppleTalk Link Access Protocol (ALAP)
Data Link Layer
Physical Layer
AppleTalk Cable

Figure 3.1.2.1 **AppleTalk protocols used by the MCC system**

3.1.2.2 Shiva Gateway

To translate from the physical LocalTalk medium to the Ethernet medium, a gateway is required. The MIPS MCC uses a FastPath gateway manufactured by Shiva Corporation. The Shiva FastPath is physically placed between the two different networks and transfers packets between them.

Configuring the Shiva Gateway is a "one time job" only. Once the Shiva Gateway is configured, it should not have to be reconfigured.

The Shiva Gateway can be configured by using the FastPath software on a Macintosh connected to the Shiva. Get a copy of the floppy entitled "Airnet Macintosh Shiva Files 10-4-92" from CM.

Copy the contents of that floppy to the Macintosh. Then, refer to the Appendix A MIPS/SHIVA Configuration and follow the directions specified in the Shiva FastPath 4 setup subsection.

Note: The Shiva is set up using phase 1 Ethernet and definitely NOT phase 2 Ethernet.

Launch Phantom before starting the SCC. To launch Phantom:

- *Log in the MIPS as Guest.*
- *Type cd /saf/bin*
- *Type ./MCC*

Shiva ID for Fort Rucker:

- The Shiva Node Number : 64
- The AppleTalk Network (between the MacIntosh and the Shiva):
  Net # : 0
  Zone : MCC-Rucker (not case sensitive)
- The IP Network (between the Shiva and the MIPS):
  Net # : 1
  Zone : n/a

Shiva ID for LORAL:

- The Shiva Node Number : 220
- The AppleTalk Network (between the MacIntosh and the Shiva):
  Net # : 320
  Zone : ADST-MCC(not case sensitive)
- The IP Network (between the Shiva and the MIPS):
  Net # : 1
  Zone : LORAL(not case sensitive)

3.1.3 Custom Hardware

There is no custom-built hardware incorporated in the MCC system. All components are "off-the-shelf" products which at one time or another were readily available from commercial sources.

3.1.4 Hardware to Software Interfaces

3.1.4.1 Macintosh Implementation

Physical communication on the AppleTalk cable is handled by the Macintosh standard hardware.

A Macintosh application has to supply buffers in advance for the ATP requests. The application is notified when an ATP request arrives by an event it encounters in its main event loop. Similarly, an application has to supply in advance a buffer for a response to a request it is issuing. An event signals the arrival of the response.

3.1.4.2 Masscomp MCC Host Implementation

The Masscomp MCC host communicates with the AppleTalk network through a Macintosh computer programmed to serve as the Bridge; The host's address (node) is simply that of the Bridge. ATSend and ATRecv are the host processes that communicate directly with the Bridge. ATP and NBP are implemented on the Masscomp MCC host as a collection of library routines where ATP routines are called to transfer ATP requests and responses, and NBP routines are called to locate network entities by name. The DDP packets are transferred via the RS-232 line. Figure 3.1.4.2 shows how these components are related.

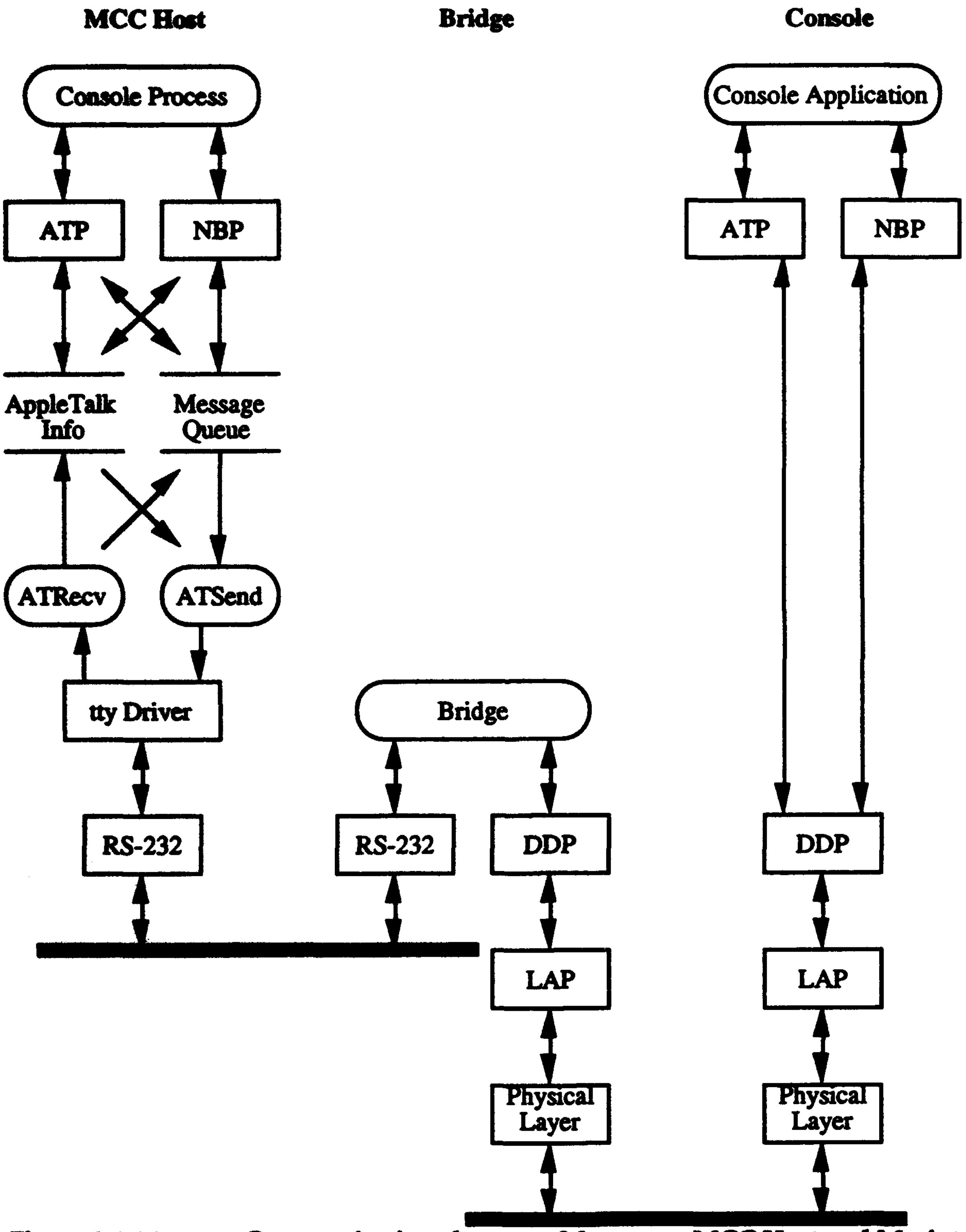


Figure 3.1.4.2 Communications between Masscomp MCC Host and Macintosh Consoles

Encoding AppleTalk ASCII packet:

- Replace any newline character with an "escape" character followed by a distinguished character such as zero.
- Replace any escape character appearing in the packet with two escape characters.
- Terminate the packet with a newline character.

Decoding AppleTalk ASCII packet:

- Replace all instances of an escape character followed by a zero with a single newline character.
- Replace all instances of two successive escape characters with a single escape character.
- Remove the terminating newline character.

To send a DDP packet from Masscomp MCC host to a Macintosh Console:

- The host places the data portion of this packet in a buffer obtained from the AppleTalk info shared memory.
- The host sends an ATalkSend message via the message queue to the ATSend process.
- ATSend encodes the source and destination addresses in the packet's header then sends it to the Bridge.
- The Bridge decodes the packet then transmits it on the AppleTalk network.

To receive a DDP packet at the Masscomp MCC host from a Macintosh Console:

- The Macintosh Bridge encodes the packet including the source and destination addresses (nodes) in the packet header then transmits the packet to the Masscomp MCC host.
- ATRecv decodes the packet and places the data portion of the packet in a buffer obtained from the AppleTalk info shared memory on the Masscomp MCC host.

3.1.4.3 MIPS MCC Host Implementation

Communication between the Macintosh MCC consoles and the MIPS MCC Simulation Host is accomplished through the use of the AppleTalk protocol and two different types of physical network media: LocalTalk and Ethernet. The MIPS MCC uses Ethernet as its network hardware interface.

To enable the two systems to communicate over the network, the CAP is used on the MIPS MCC. CAP gives the MIPS MCC the ablity to "speak" the AppleTalk protocols reliably over the unreliable User Datagram Protocol (UDP).

To translate from the physical LocalTalk medium to the Ethernet medium, a gateway is required. The MIPS MCC uses a COTS FastPath gateway manufactured by Shiva Corporation; all packets sent between the MCC consoles and the MIPS MCC pass through this gateway. The Shiva FastPath is physically placed between the two different networks and transfers packets between them.

Messages from the MCC consoles to the MIPS MCC are sent via the LocalTalk hardware interface to the FastPath using the AppleTalk protocol. The FastPath gateway repackages the messages as AppleTalk over UDP, and transmit them to the MIPS MCC over Ethernet. The MIPS MCC receives the messages and passes them to the CAP software for decoding and delivery to the MIPS MCC application. Figure 3.1.4.3 shows the setup and the relationship between the communication components.

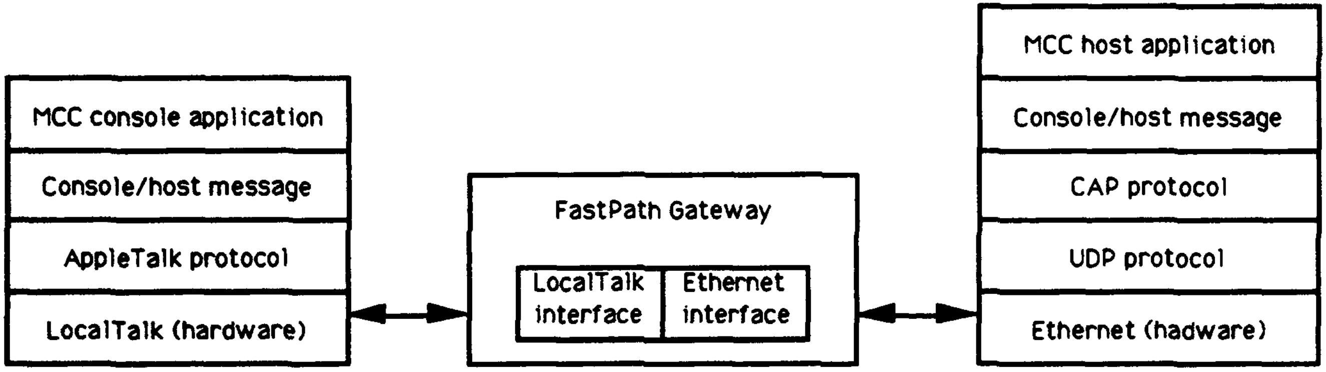


Figure 3.1.4.3 Communication via FastPath gateway

4 Software Description

4.1 MIPS MCC Host Software Description

4.1.1 Overview of MIPS MCC Host Software

The MIPS MCC host software is written in C and is a modification of the MIPS SAF 4.0.4 code. The main process that runs on the MIPS MCC host software is called phantom and its code resides in the file main.c under the /saf/src/host/ directory. All of the data (.lisp) files for the MIPS MCC host software reside in the directory /saf/config. Communications with the SCC console is done using the Ethernet UDP protocol across the SIMNET network.

4.1.2 Development Environment Description

4.1.2.1 Development Environment Directory Structure

The development environment for the MIPS MCC host software is identical in structure to the MIPS SAF 4.0.4 software. The development environment has a top directory called saf. Below the saf directory are the following directories:

common
contains all common libraries and include files (similar to common directories on other SIMNET models, but many of the libraries are unique to the MIPS MCC host).

config
contains all data (.lisp) files used by the MIPS MCC host and also by the SCC console via aufs which is a file sharing application that is part of the MIPS MCC host software.

include
contains directories containing include files local to the MIPS MCC host software.

lib
contains local library objects.

libsrc
contains directories of source code for each local library.

src
contains host directory which contains all of the source code local to the MIPS MCC host software. Note that there is also a config directory here that contains the same data files in /saf/config. The files in /saf/config are considered the baseline data files and the config directory under /saf/src/host/ should be ignored.

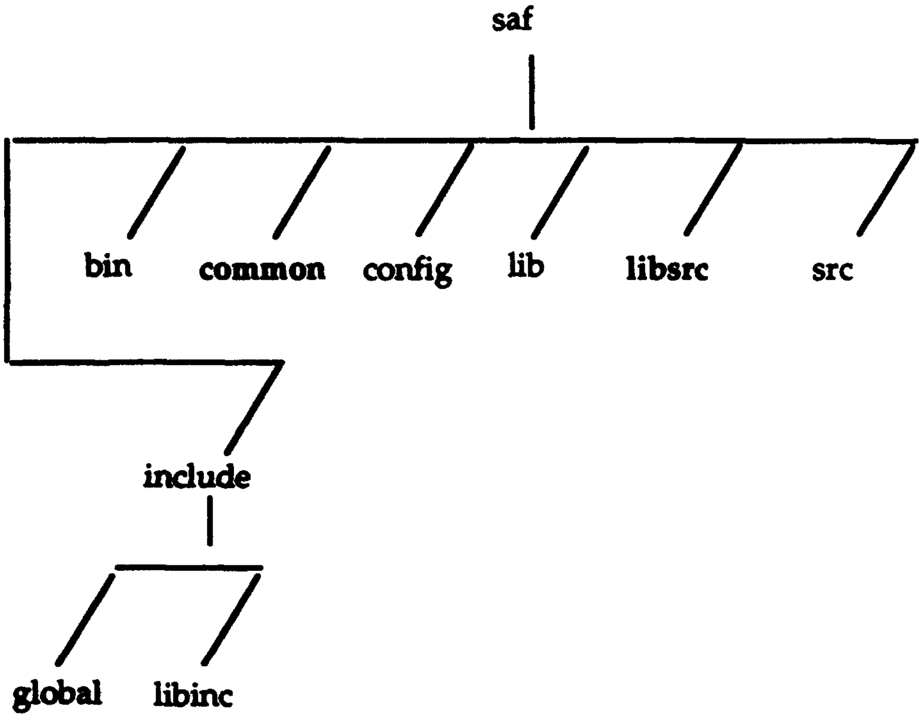


Figure 4.1.2.1 MIPS MCC /saf directory structure

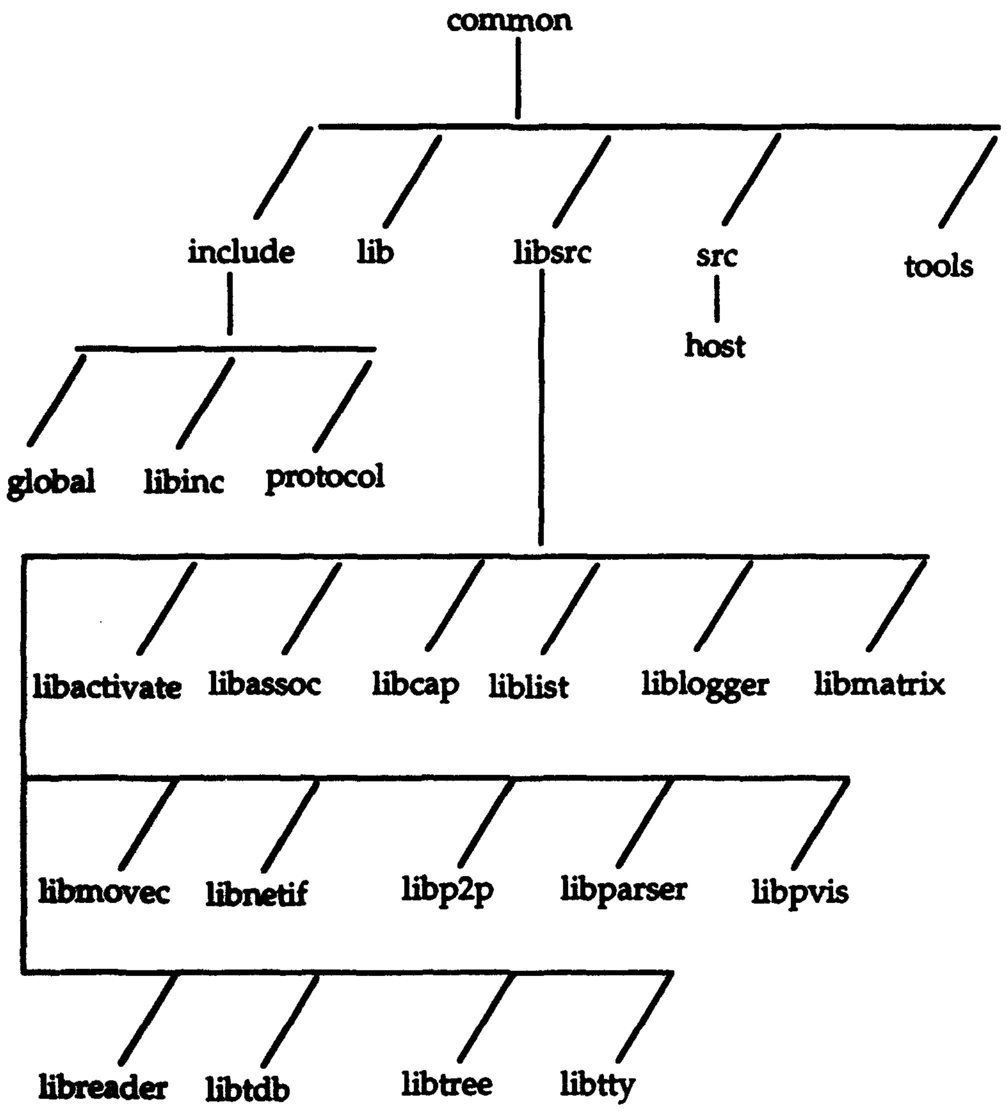


Figure 4.1.2.2 MIPS MCC /saf/common directory structure

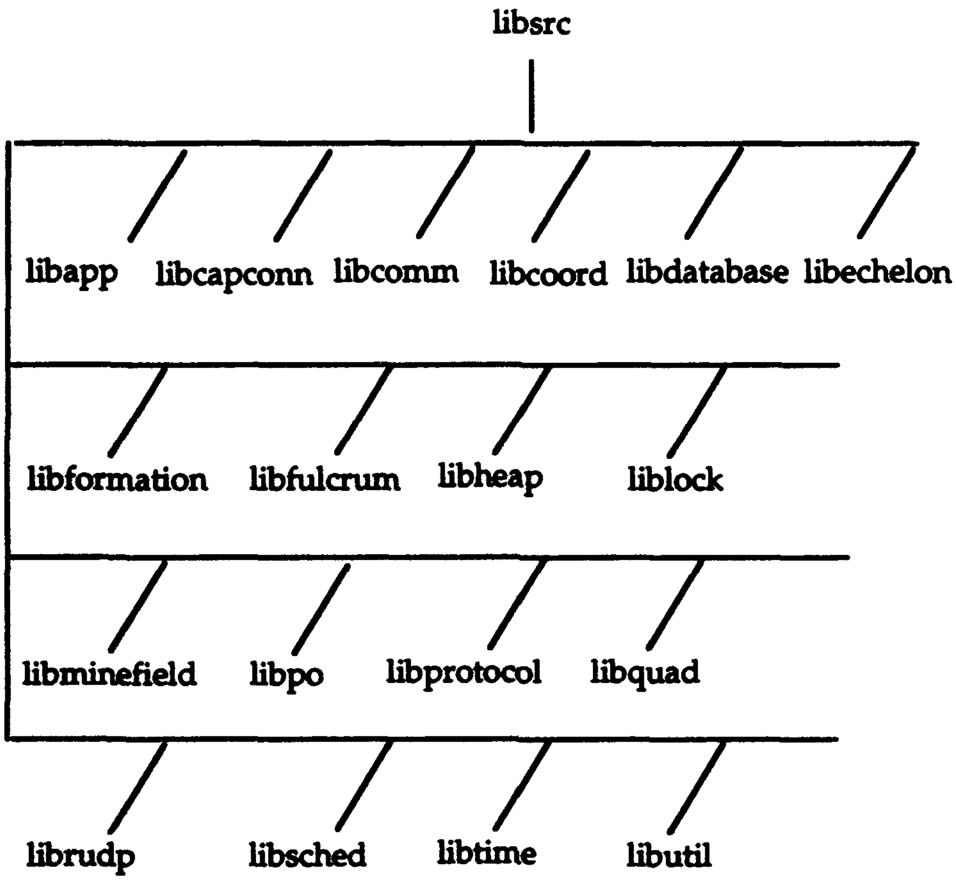


Figure 4.1.2.3 MIPS MCC /saf/libsrc directory structure

4.1.2.2 Development Environment Build Configuration

Most configuration choices with the MIPS MCC host software are made by data file changes and through parameters on the phantom application command line interface.

4.1.3 Runtime Environment Description

The MIPS MCC host software resides entirely on a MIPS 2000 or 3000. Along with the phantom process the MIPS MCC host software must have processes running for the Ethernet and the CAP (Columbia AppleTalk Protocol) support. The CAP software is freeware software obtained from the Internet network and implemented in the MIPS MCC host software. The CAP software is integrated with the MIPS MCC host software and is not considered a separate piece of software when used in the MIPS MCC host. The process that supports the Ethernet communications on the MIPS is called "ringstart". The processes that support the CAPs software for file sharing between UNIX and Macintosh systems are aufs and atis along with a daemon called atalkad. The executables for these processes are located at /usr/local/lib on the MIPS MCC host. These aufs (Apple Unix File Sharing) processes need the AFPVOLS, atalk.local, and atalkatab configuration files in the /usr/local/lib directory.

4.1.3.1 Runtime Environment Directory Structure

All of the directories used by the MIPS MCC host are located in /saf and /usr/local/lib Important directories under the /saf directory are:

bin
contains all executables including Ethernet drivers and phantom (main process).
config
contains all data files used by the MIPS MCC host and the Macintosh SCC.
terrain
contains all terrain database information used by the MIPS MCC host

The /usr/local/lib directory contains the following files: AFPVOLS, atis, aufs.

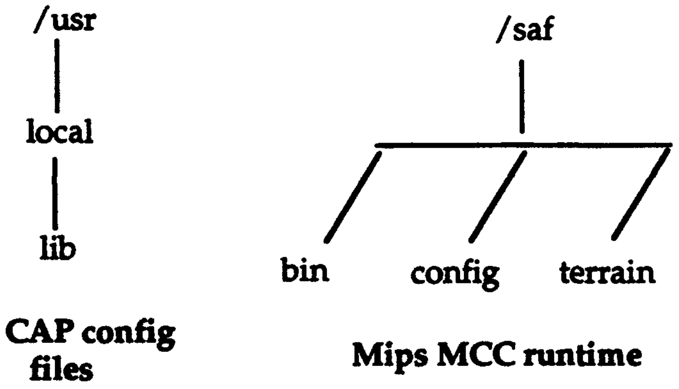


Figure 4.1.3.1 MIPS MCC Runtime Environment directory structure

4.1.3.2 Runtime Environment Data Files

The data files used by the MIPS MCC host are located in directories /saf and /usr/local/lib. The /saf directory contains all of the .lisp files used by the MIPS MCC host software. A detailed description of many of the .lisp files is in The AIRNET SDD Volume II: Comanche Upgrades. The data files in the directory /usr/local/lib contain definitions for the connection between the MIPS MCC host and the Shiva fastpath gateway between the SIMNET network and the Macintosh 1.0.0 SCC.

These files are :

AFPVOLS
: defines mapping from MIPS Unix directory name to the Macintosh folder name that appears on the Chooser on the Macintosh.

atalk.local
: defines Internet to AppleTalk mappings

atalkatab
: Database of Internet and AppleTalk definitions used by the CAP aufs software.

4.1.3.3 Runtime Environment Command Arguments

Below is a list of the command line arguments available when launching the phantom process. This list will be displayed if you launch the phantom nrocess with the -h command line argument.

```
****************************************************************************
* PHANTOM 4.0.4-MCC 7.2.0, Released on 6/09/92                                *
****************************************************************************
Switches:
 -c(ache terrain to speed startup)
 -d(ebugging on)
 -e(xercise id) #
 -i(solate connection) #
 -n(o network packets sent)
 -p(hantom commands debugging on)
 -s(leep occasionally to allow workstation on same machine)
 -t(errain database to use) filename
 -v(ehicle projector mode)
 -? or -h (print help)
 -6(version 6.0 simnet constants)
 -2(Ethernet 2 packets [not 802.3])
 -z(Restore the terminal after a program crash, and exit)
 -o(record shots to file firedata.  follow with 'red', 'blue', or 'all')
```

4.1.3.4 Runtime User Interface Commands

The MIPS MCC host has a relatively robust keyboard interface. Typing the question mark key (?) at any time will give you a list of command or parameter options available to the user.

4.2 AIRNET SCC Software Description

4.2.1 Overview of AIRNET Macintosh SCC Software

The AIRNET Macintosh SCC software resides on a Macintosh running at least 6.0.5 version of the Macintosh operating system. The AIRNET Macintosh SCC application was designed and built with ThinkC 5.0. The AIRNET Macintosh SCC takes advantage of the ThinkC class libraries provided by version 5.0 making the code more object oriented than code produced with ThinkC 4.0.

4.2.2 Development Environment Description

4.2.2.1 Development Environment Directory Structure

The development environment of the AIRNET Macintosh SCC is on a Macintosh under the ThinkC development folder. At the top level of the SCC folder are all the projects, libraries, and four folders. The four folders are named Common, InitMac, Libraries Folder, and SCC. The Common folder contains ThinkC class definitions that are common throughout all sections of the SCC source code. The InitMac folder contains only one source file with one small function that performs initialization of the Macintosh Toolbox which provides support for the user interface on the Macintosh.

4.2.2.2 Development Environment Build Configuration

The only important files used to affect the build configuration of the AIRNET Macintosh SCC application are the SCC project file and the resource file SCC.π.rsrc which is linked into the SCC application by ThinkC when you are building the actual application.

4.2.3 Runtime Environment Description

The directory in which the SCC application resides can be any user directory. The user of the SCC application must make sure that the /saf/config filesystem on the MIPS MCC host is remotely mounted on the AIRNET Macintosh SCC as a filesystem named MCC. The filesystem is mounted on the AIRNET Macintosh SCC through the Chooser with Appleshare.

4.2.3.1 Runtime Environment Directory Structure

The SCC application can reside on any user directory and still read the data files needed.

4.2.3.2 Runtime Environment Data Files

The data files needed reside in a remotely mounted filesystem on the MIPS.

4.2.3.3 Runtime Environment Command Arguments

There are no command arguments in a normal Macintosh application.

4.2.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user inteface commands available to the user.

4.3 Masscomp MCC Host Software Description

4.3.1 Overview of Masscomp MCC Host Software

The Masscomp MCC Host Software runs on a Masscomp 5600 computer with a CMC ENP30 Ethernet card and four RS-232 ports, one of which is connected to a Macintosh computer that acts as a bridge between the RS-232 communications from the Masscomp MCC host and the AppleTalk communications used by the Macintosh Consoles. There are several processes that run on the Masscomp MCC host. The main process is called Mother and is responsible for initialization of the MCC software and Ethernet communications. There are processes running for each Macintosh console. These processes are SCC, Admin, Maint, FSE, CAS , CEC, and Place (NOTE : The Macintosh Place console is not used) and send messages between process on the Masscomp via IPC's (Interprocess Communications). Detailed information on IPC's can be obtained from the Masscomp system documentation. There also exists two other processes called ATSend and ATRecv which are responsible for sending and receiving Appletalk messages between the Masscomp MCC Host and the Macintosh consoles.

4.3.2 Development Environment Description

4.3.2.1 Development Environment Directory Structure

The development environment directory structure of the Masscomp MCC Host has the following directories at the top level common, include, lib, libsrc, src, and tools. The common directory is similar to the common directories on other SIMNET simulators with common libraries and include files. The include directory is empty. The lib directory is where produced library (.a) files will be placed. The libsrc directory is empty. The src directory contains one directory called mcc which contains the local code to build all of the MCC processes including the local libraries libmcc and libcli.

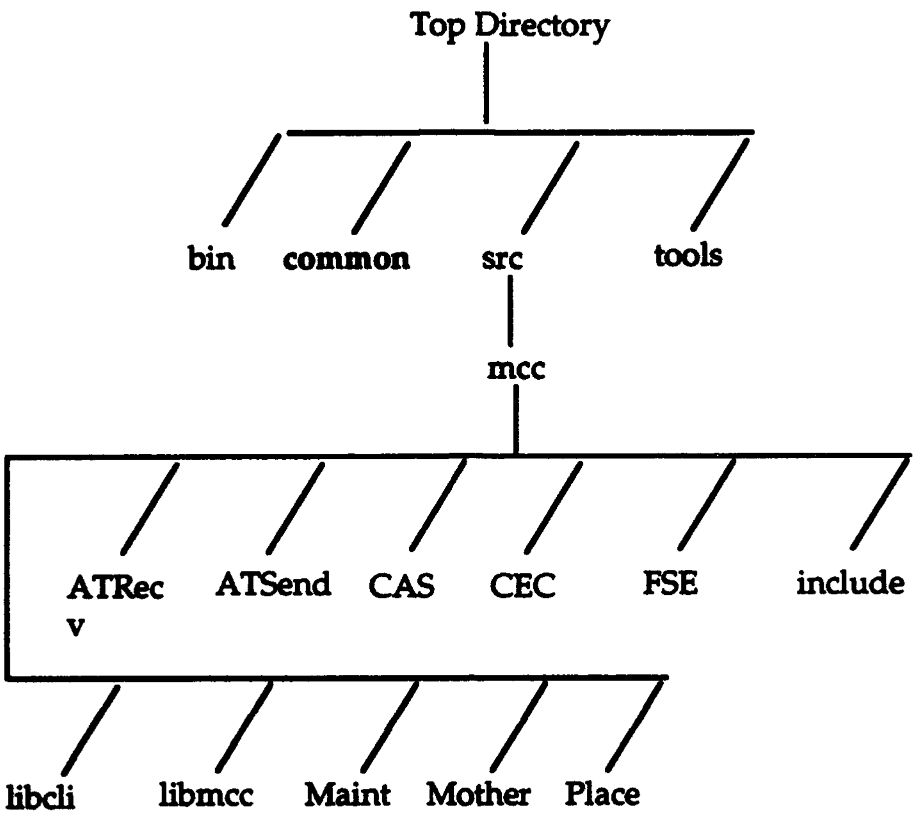


Figure 4.3.2.1 Masscomp MCC Top directory structure

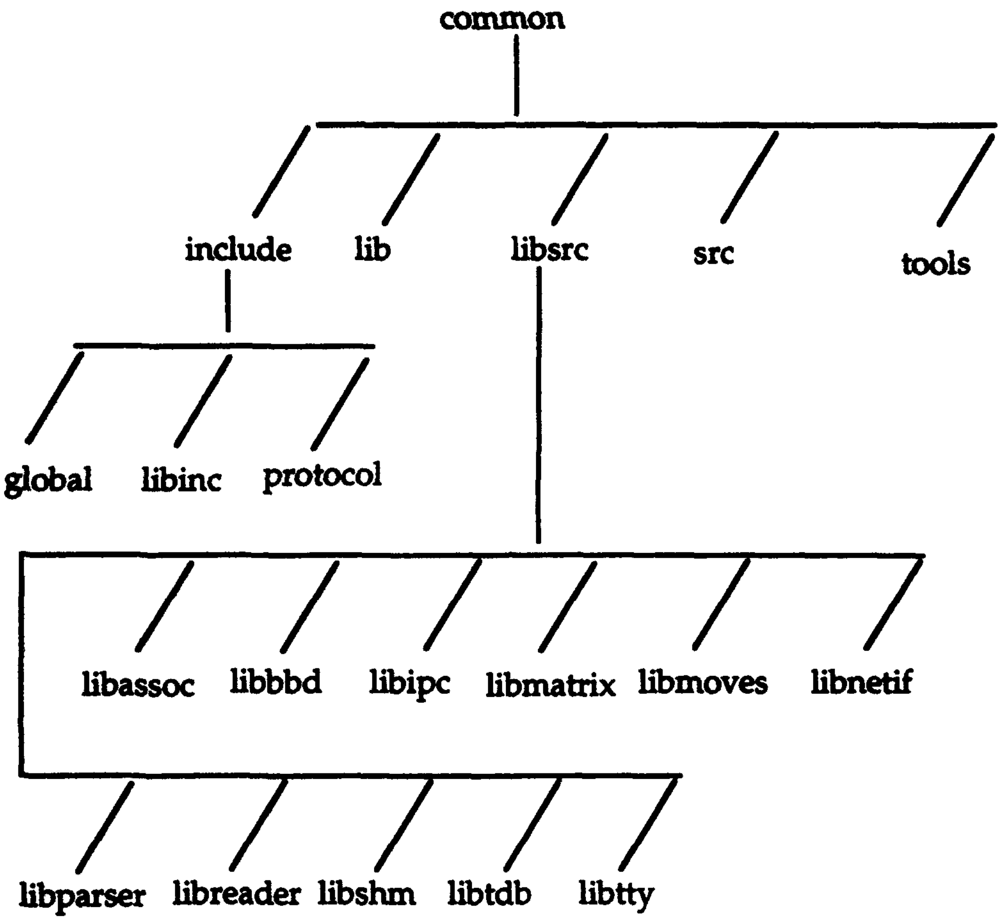


Figure 4.3.2.2 **Masscomp MCC common directory structure**

4.3.2.2 Development Environment Build Configuration

The build configuration of the MCC can be modified so that the processes are debuggable or not. Processes and Executables produced with debug options set on should not run any slower, but will have considerably larger execution sizes. If you want to debug the code then you must set the debug option (-g) in the make variable CCFLAGS.

4.3.3 Runtime Environment Description

4.3.3.1 Runtime Environment Directory Structure

The directory structure of the Masscomp MCC runtime environment consists of the following directories :

- /simnet/bin

  This directory contains the following files which form the Masscomp MCC host executable.

  - MCC - Unix shell script to start up the processes below
  - MCC-Mother - Mother process
  - MCC-SCC - SCC process
  - MCC-Admin - Admin process
  - MCC-Maint - Maint process
  - MCC-FSE - FSE process
  - MCC-CAS - CAS process
  - MCC-CEC - CEC process
  - MCC-Place - Place process
- /simnet/data

  This directory contains the following data files

  Files unique to MCC

  - Knox.0311 - Terrain database used by MCC located at /simnet/terrain/data/knox-0311 and is close to 30 MB in size.
  - MCC-pars - Text file for reconfigurable portions of MCC Host.
  - MCC-pars.alt - continuation of MCC-pars.
  - ExerciseLog - Output from MCC Host.
  - VehicleLog - Output from MCC Host.

  Files common to most SIMNET hosts

  - assoc.def - text file to define host information to the SIMNET network.
  - network.def - more SIMNET information.
  - simnet.mac - defines SIMNET object type values.

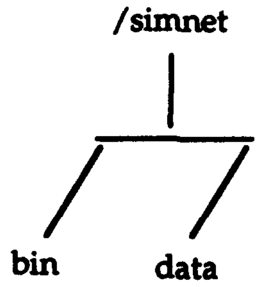


Figure 4.3.3.1 Masscomp MCC Runtime Directory Structure

4.3.3.2 Runtime Environment Data Files

See 4.3.3.1 Runtime Environment Directory Structure.

4.3.3.3 Runtime Environment Command Arguments

MCC mother is the main process and it is passed the name of the data file to read (MCC-pars) which references MCC-pars.alt (the other data file that has to be read in). The terrain data base files available to the MCC software are listed in the data file MCC-pars and are only selectable from the SCC Initialization screens on the Macintosh SCC console.

4.3.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user inteface commands available to the user.

4.4 Macintosh SCC Software Description

4.4.1 Overview of Macintosh SCC Software

The Macintosh SCC Software is the largest and most complex of all the Macintosh Consoles and comunicates with the Masscomp MCC and all of the other Macintosh consoles.

4.4.2 Development Environment Description

4.4.2.1 Development Environment Directory Structure

The directory structure of the development environment resides under the ThinkC development folder on the Macintosh. There are three library folders called libmac, libsim, and libsupply. There are two include folders called General Includes and include. There is the folder for the SCC appplication called SCC, which contains source files and local include files.

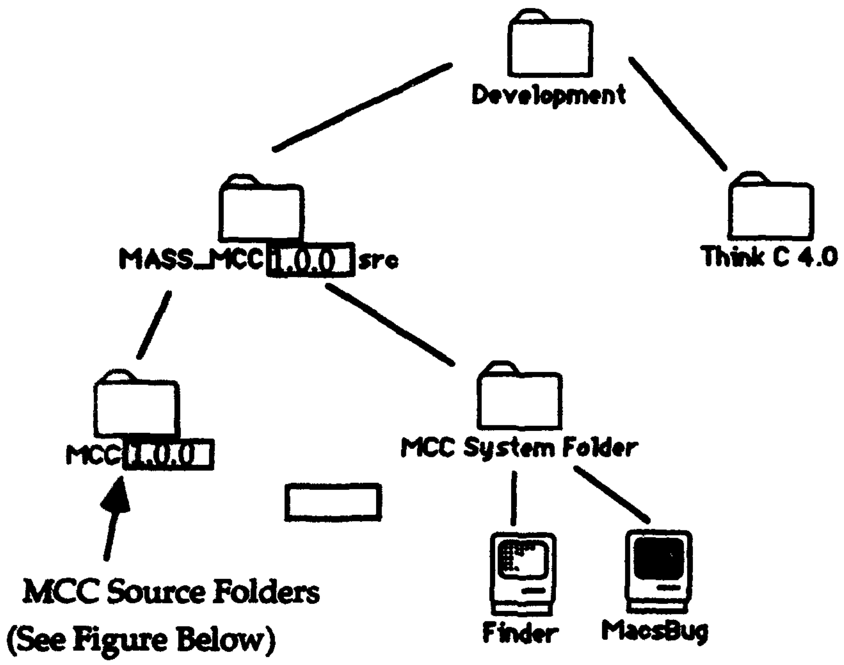


Figure 4.4.2.1 Top Level Macintosh Development Environment

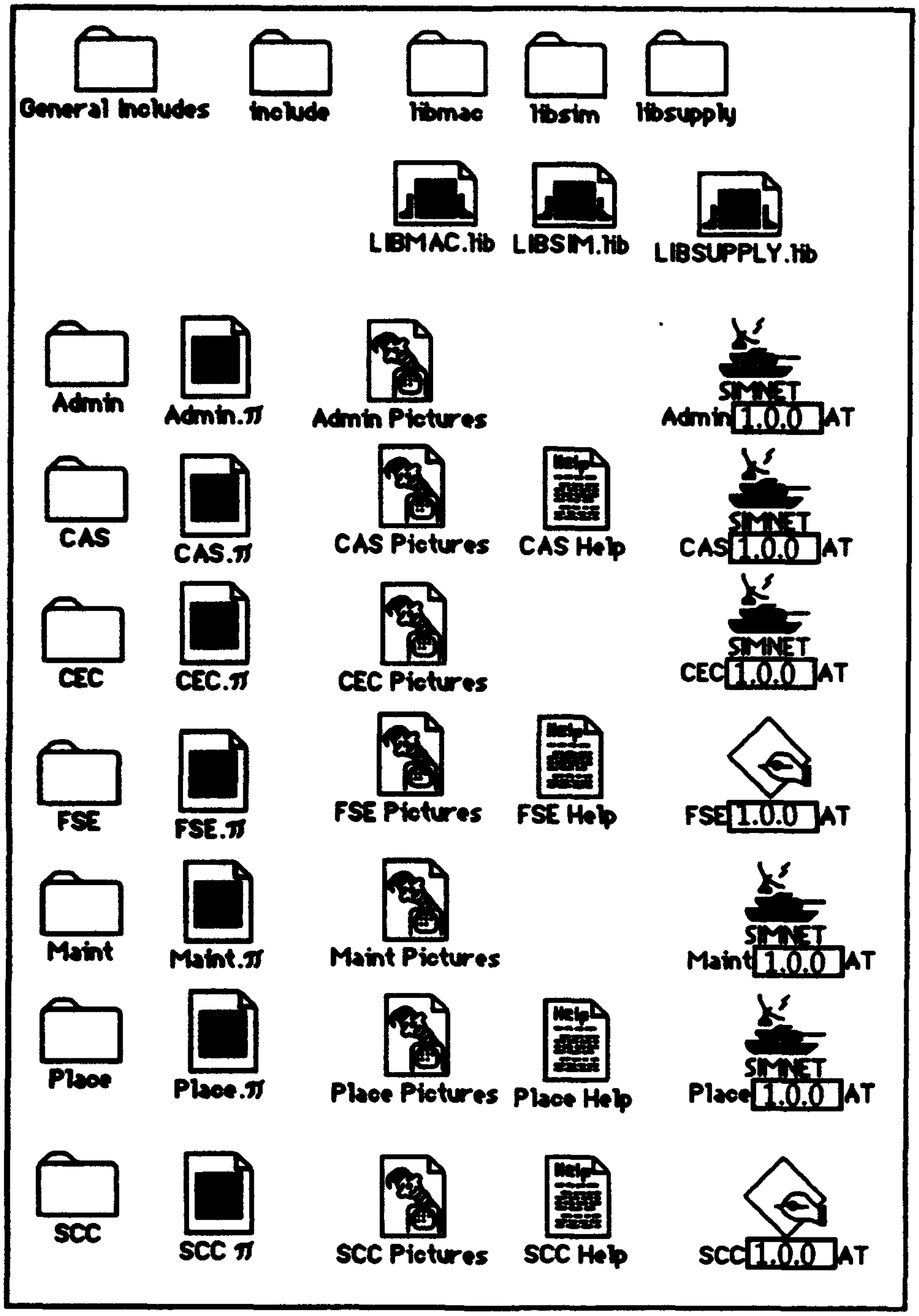


Figure 4.4.2.2 MCC Macintosh Development Environment

4.4.2.2 Development Environment Build Configuration

The SCC application can be built to run in AppleTalk or standalone versions. The build version is determined by the definition of VERSION in the file version_scc.h. If VERSION is equal to APPLETALK then the SCC will run in normal operation, if the VERSION is equal to FULDAGAP then the SCC will run in standalone mode using canned data.

4.4.3 Runtime Environment Description

4.4.3.1 Runtime Environment Directory Structure

The SCC executable can be run in any normal folder on the Macintosh. Files other than the SCC executables needed for running are Sim Pictures, SIMNET Pictures, SCC Pictures and SCC Help.

4.4.3.2 Runtime Environment Data Files

No data files are used by this Macintosh console.

4.4.3.3 Runtime Environment Command Arguments

There are no command arguments in a normal Macintosh application.

4.4.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user interface commands available to the user.

4.5 Macintosh Admin Console Software Description

4.5.1 Overview of Macintosh Admin Console Software

The Macintosh Maint application provides tables of resupply trucks (fuel, ammo, pallet of mines) that allow the user to dispatch resupply trucks in an exercise. The Admin Console is initialized along with the Maint Console via the Battlemaster during initialization of the CSS.

4.5.2 Development Environment Description

4.5.2.1 Development Environment Directory Structure

The development environment for the Admin application is the same directory structure used by the SCC application where there is also a directory called Admin for the local code on the Admin application.

4.5.2.2 Development Environment Build Configuration

The Admin application can be built to run in AppleTalk or standalone versions. The version of the application is determined by the definition of VERSION in the file version_admin.h. If VERSION is equal to APPLETALK then the Admin will run in normal operation, if the VERSION is equal to STANDALONE then the Admin will run in standalone mode using canned data.

4.5.3 Runtime Environment Description

4.5.3.1 Runtime Environment Directory Structure

The Admin executable can be run in any normal folder on the Macintosh. Files other than the Admin executables needed for running are Sim Pictures, SIMNET Pictures, and Admin Pictures.

4.5.3.2 Runtime Environment Data Files

No data files are used by this Macintosh console.

4.5.3.3 Runtime Environment Command Arguments

There are no command arguments in a normal Macintosh application.

4.5.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user interface commands available to the user.

4.6 Macintosh Maint Console Software Description

4.6.1 Overview of Macintosh Maint Console Software

The Macintosh Maint application provides tables of maintenance trucks that allow the user to dispatch maintenance trucks in an exercise. The Maint Console is initialized along with the Admin Console via the Battlemaster during initialization of the CSS.

4.6.2 Development Environment Description

4.6.2.1 Development Environment Directory Structure

The development environment for the Maint application is the same directory structure used by the SCC application where there is also a directory called Maint for the local code on the Maint application.

4.6.2.2 Development Environment Build Configuration

The Maint application can be built to run in AppleTalk or standalone versions. The version of the application is determined by the definition of VERSION in the file version_maint.h. If VERSION is equal to APPLETALK then the Maint will run in normal operation, if the VERSION is equal to STANDALONE then the Maint will run in standalone mode using canned data.

4.6.3 Runtime Environment Description

4.6.3.1 Runtime Environment Directory Structure

The Maint executable can be run in any normal folder on the Macintosh. Files other than the Maint executables needed for running are Sim Pictures, SIMNET Pictures, and Maint Pictures.

4.6.3.2 Runtime Environment Data Files

No data files are used by this Macintosh console.

4.6.3.3 Runtime Environment Command Arguments

There are no command arguments in a normal Macintosh application.

4.6.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user interface commands available to the user.

4.7 Macintosh FSE Console Software Description

4.7.1 Overview of Macintosh FSE Console Software

The Macintosh FSE software provides screens and dialogs to perform control the fire of Mortar and Artillery Batteries. The FSE is initialized via the Battlemaster on the Macintosh SCC Console.

4.7.2 Development Environment Description

4.7.2.1 Development Environment Directory Structure

The development environment for the FSE application is the same directory structure used by the SCC application where there is also a directory called FSE for the local code on the FSE application.

4.7.2.2 Development Environment Build Configuration

The FSE application can be built to run in AppleTalk or standalone versions. The version of the application is determined by the definition of VERSION in the file version_fse.h. If VERSION is equal to APPLETALK then the FSE will run in normal operation, if the VERSION is equal to FULDAGAP then the FSE will run in standalone mode using canned data.

4.7.3 Runtime Environment Description

4.7.3.1 Runtime Environment Directory Structure

The FSE executable can be run in any normal folder on the Macintosh. Files other than the FSE executables needed for running are Sim Pictures, SIMNET Pictures, FSE Pictures, and FSE Help.

4.7.3.2 Runtime Environment Data Files

No data files are used by this Macintosh console.

4.7.3.3 Runtime Environment Command Arguments

There are no command arguments in a normal Macintosh application.

4.7.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user interface commands available to the user.

4.8 Macintosh CAS Console Software Description

4.8.1 Overview of Macintosh CAS Console Software

The Macintosh CAS software provides screens and dialogs to perform Preplanned and On Call Close Air Support missions. The CAS is initialized via the Battlemaster on the Macintosh SCC Console.

4.8.2 Development Environment Description

4.8.2.1 Development Environment Directory Structure

The development environment for the CAS application is the same directory structure used by the SCC application where there is also a directory called CAS for the local code on the CAS application.

4.8.2.2 Development Environment Build Configuration

The CAS application can be built to run in AppleTalk or standalone versions. The version of the application is determined by the definition of VERSION in the file version_cas.h. If VERSION is equal to APPLETALK then the CAS will run in normal operation, if the VERSION is equal to FULDAGAP then the CAS will run in standalone mode using canned data.

4.8.3 Runtime Environment Description

4.8.3.1 Runtime Environment Directory Structure

The CAS executable can be run in any normal folder on the Macintosh. Files other than the CAS executables needed for running are Sim Pictures, SIMNET Pictures, CAS Pictures, and CAS Help.

4.8.3.2 Runtime Environment Data Files

No data files are used by this Macintosh console.

4.8.3.3 Runtime Environment Command Arguments

There are no command arguments in a normal Macintosh application.

4.8.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user interface commands available to the user.

4.9 Macintosh CEC Console Software Description

4.9.1 Overview of Macintosh CEC Console Software

The Macintosh CEC software provides screens and dialogs to perform Emplace, Breach and Move minefield operations. The CEC is initialized via the Battlemaster on the Macintosh SCC Console.

4.9.2 Development Environment Description

4.9.2.1 Development Environment Directory Structure

The development environment for the CEC application is the same directory structure used by the SCC application where there is also a directory called CEC for the local code on the CEC application.

4.9.2.2 Development Environment Build Configuration

The CEC application can be built to run in AppleTalk or standalone versions. The version of the application is determined by the definition of VERSION in the file version_fse.h. If VERSION is equal to APPLETALK then the CEC will run in normal operation, if the VERSION is equal to FULDAGAP then the CEC will run in standalone mode using canned data.

4.9.3 Runtime Environment Description

4.9.3.1 Runtime Environment Directory Structure

The CEC executable can be run in any normal folder on the Macintosh. Files other than the CEC executables needed for running are Sim Pictures, SIMNET Pictures and CEC Pictures.

4.9.3.2 Runtime Environment Data Files

No data files are used by this Macintosh console.

4.9.3.3 Runtime Environment Command Arguments

There are no command arguments in a normal Macintosh application.

4.9.3.4 Runtime User Interface Commands

See the ADST MCC Operations Manual for a description of the user interface commands available to the user.

5. Notes

5.1 Acronyms:

ALAP	AppleTalk Link Access Protocol
ATP	AppleTalk Transaction Protocol
AUFS	Apple Unix File Sharing
CAP	Columbia AppleTalk Protocol
CAS	Close Air Support
CEC	Combat Engineer Console
CMC	Communication Machinery Corporation
COMANCHE	RAH-66 Helicopter
CPU	Central Processing Unit
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSS	Combat Service Support
CSU	Computer Software Unit
DDP	Data Delivery Protocol
FSE	Fire Support Element
HPSM	High Performance Serial Multiplexor
IPC	Interprocess Communications
MCC	Management Command and Control System
NBP	Name Binding Protocol
PDU	Protocol Data Unit
RAH-66	Comanche helicopter
RWA	Rotary Wing Aircraft
SCC	SIMNET Control Console
SDD	Software Design Document
SIMNET	SIMulation NETwork
TOC	Tactical Operations Center
UDP	User Datagram Protocol

5.2 Default Ammo Truck Loads

The following tables detail the default ammunition loads for Soviet and United States forces:

5.2.1 Default US Ammo Load for Ammo Truck

Truck Number(s)	Munition Name	Amount
1-6	ammoHEAT105	190
1-6	ammoAPDS105	122
7-8	ammoHEI25	40
7-8	ammoAPDS25	40
7-8	ammoMissileTOW	13
7-8	ammoDRAGON	26
9	ammoHEI25	1
9	ammoHEAT105	180
9	ammoAPDS25	11
9	ammoAPDS105	60
10	ammoHEI25	40
10	ammoAPDS25	40
10	ammoMissileTOW	30

5.2.2 Default USSR Ammo Load for Ammo Trucks

Truck Number(s)	Munition Name	Amount
1-6	ammoHEAT105	50
1-6	ammoAPDS105	50
7-8	ammoHEI25	20
7-8	ammoAPDS25	20
7-8	ammoMissileTOW	3
7-8	ammoMissileFAAD	3
9	ammoHEI25	1
9	ammoHEAT105	50
9	ammoAPDS105	20
10	ammoAPDS25	20
10	ammoHellfire	15

5.2.3 Default US or USSR Ammo Load for Pallet Trucks

Truck Number(s)	Munition Name	Amount
1-5	ammoATScatMine	70
1-5	ammoAPBScatMine	50
1-5	ammo APFScatMine	50
6-10	ammoATConvMine	120
6-10	ammoAPBConvMine	60
6-10	ammoAPFConvMine	60

5.3 Munitions Nomenclature

The following table details the nomenclatures, weights, and number of rounds carried by a RAH-66 Comanche armed reconnaissance helicopter.

Munition Designator	20mm HEI	20 mm PIE
Munition Name	M56	PGU28
Type of Munition	High Explosive Incindiary	Pyrotechnically Initiated Explosive
Munition Weight	0.726 lbs	0.726 lbs
Rounds per Box	100	100
Ammo Box Dimensions	18.59" x 14.59" x 8.3"	
Ammo Box Volume	1.3 Cubic Feet	1.3 Cubic Feet
Damage Tables	See Below	See Below

5.4 Files changed for the MASSCOMP MCC Comanche Upgrade

The following files contain changes to implement Comanche Compatibility in the MASSCOMP-based AIRNET Management Command, and Control system.

```
common/include/protocol/mun_type.h

src/mcc/MCC-pars.alt

src/mcc/Admin/displace.c

src/mcc/Admin/main.c

src/mcc/Admin/service.c

src/mcc/SCC/css.c

src/mcc/include/ammo.h

src/mcc/include/MCC_limits.h
```

APPENDIX A - MIPS/SHIVA Configuration

1. MIPS Setup

To "Launch" the 3 Daemons :

Make sure "/etc/init.d/netdaemons" (code that inits deamons) includes the following code:

```
if test -x /etc/atalkad; then
   /etc/atalkad;  echo "atalkad\c"
fi

if test -x /usr/local/lib/atis; then
   /usr/local/lib/atis;   echo "atis\c"
fi

if test -x /usr/local/lib/aufs; then
   /usr/local/lib/aufs -G guest -V /usr/local/lib/AFPVOLS; echo "aufs\c"
fi
```

Misc

CAP - Columbia AppleTalk Package for UNIX.

The code is located in /usr3/cap60. "CAP" is used in it's traditional config. (IPTalk mode).

Daemons The Must Run

atalkad
(apple talk administrator)

directory /etc

sum atalkad = 14369 486
(checksum)

atis
(apple talk interconnect system)

directory /usr/local/lib

sum atis = 58137 615
(checksum)

aufs
(apple unix file server)
= this is a CAP application =

directory /usr/local/lib

[aufs works as a file server for Macintosh computers with Appleshare client code]

sum aufs = 11724 1020
(checksum)

File Information

/etc/hosts (used to assign IP address to logical names)

```
127.1             local host
192.67.225.201    MIPS-1
192.67.225.10     ADST-MCC          * identifies SHIVA and the Appletalk net
```

/etc/local_hostname (this is where we set name of host)

```
MIPS-1
netmask   0xffffff00   broadcast   192.67.225.0
```

/etc/passwd (we have to have a "guest" entry for the MAC)

```
guest::1300:1:Guest Account:/usr/local/lib:/bin/csh          (1300 is the user number)
```

/usr/local/lib/AFPVOLS (tells MAC which volume to load)

```
/usr/saf/config:MCC:
```

/saf/bin/assoc.def

```
site 1        (these numbers are ordinarily
host 11       assigned by BBN - we arbitrarily
              assigned them here.
```

When editing this file, DO NOT put a carriage return after the host number! Make sure the host number is at the end of the file!!!!!

/etc/atalk.local (contains routing info for the static IPTalk)

```
appletalk net    node    zone                    zone name MUST be zone name
_____________________________________            of applenet THIS Mips is on.
    0.1          201     LORAL  <----------/
    0.1          10       192.67.225.10
```

/saf/bin/enp.bin (this is the board code for Ethernet)

This is the board code for the ethernet port

/saf/config.factory/"machine.lisp"

```
CHANTYPES (0)
UDP (1)
CAP (2)
(
 (PORTS  CHANTYPES
  (port_type  CAP 3))
  (ethernet_device "/dev/enp0")
)
```

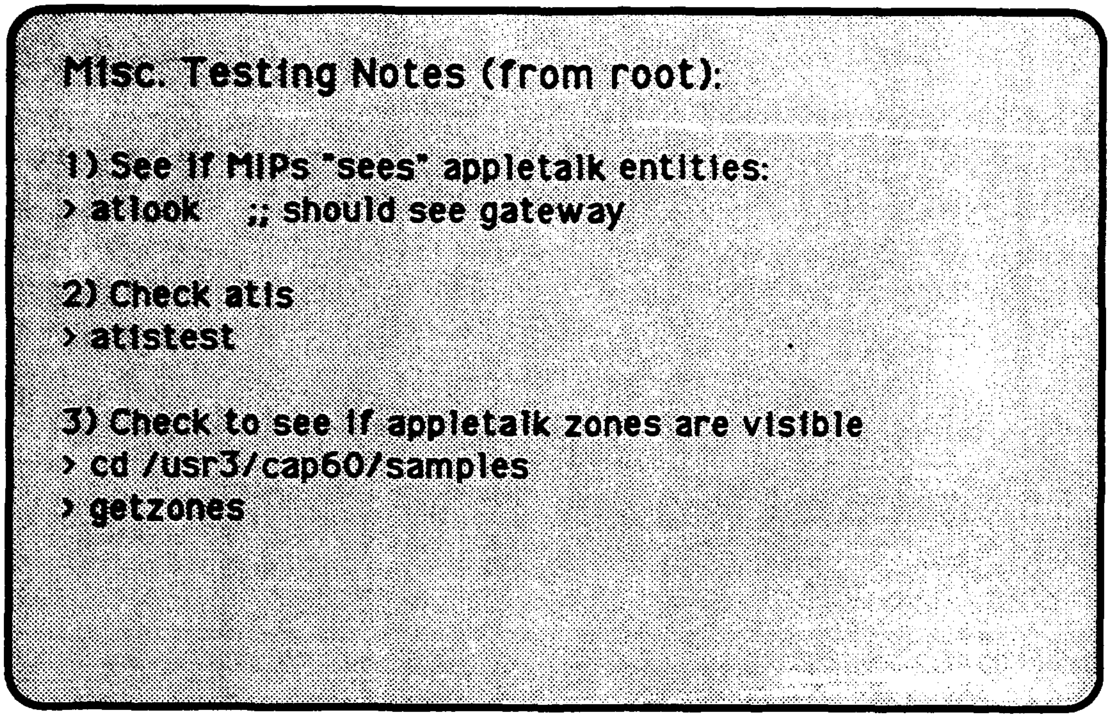
/etc/atalkatab (def. mappings for IP subnets to AppleTalk Network)

```
0.1       N1     192.67.225.0      LORAL
1.64      K      192.67.225.10     ADST-MCC
 1192.67.225.255   LO        *four byte IP address
 LO LO                        *ipdebug  ipfile
 LO LO LO LO    SO SO        *ipother  unused   unused
 LXO          SO    SO       *flags   ipstatic  ipdynamic
 %N    %N                    *atneta   atnete
```

1.64 = applenet 320
((1*256) + 64 = 320)
K - Anet is the atalk port on SHIVA
C - SHIVA is also "core gateway"

Describes "configuration info" for SHIVA

This line describes an appletalk net ("anet")
0.1 - anet number: This MIPs is on applenet 1
N1 - flags: anet equivalent to IP net, 1 means broadcast address of 0.0.0.255 ANDed with IP address
192.67.225.0 - IP address
LORAL - zone name



Misc. Testing Notes (from root):

1) See if MIPs "sees" appletalk entities:

```
> atlook   ;; should see gateway
```

2) Check atis

```
> atistest
```

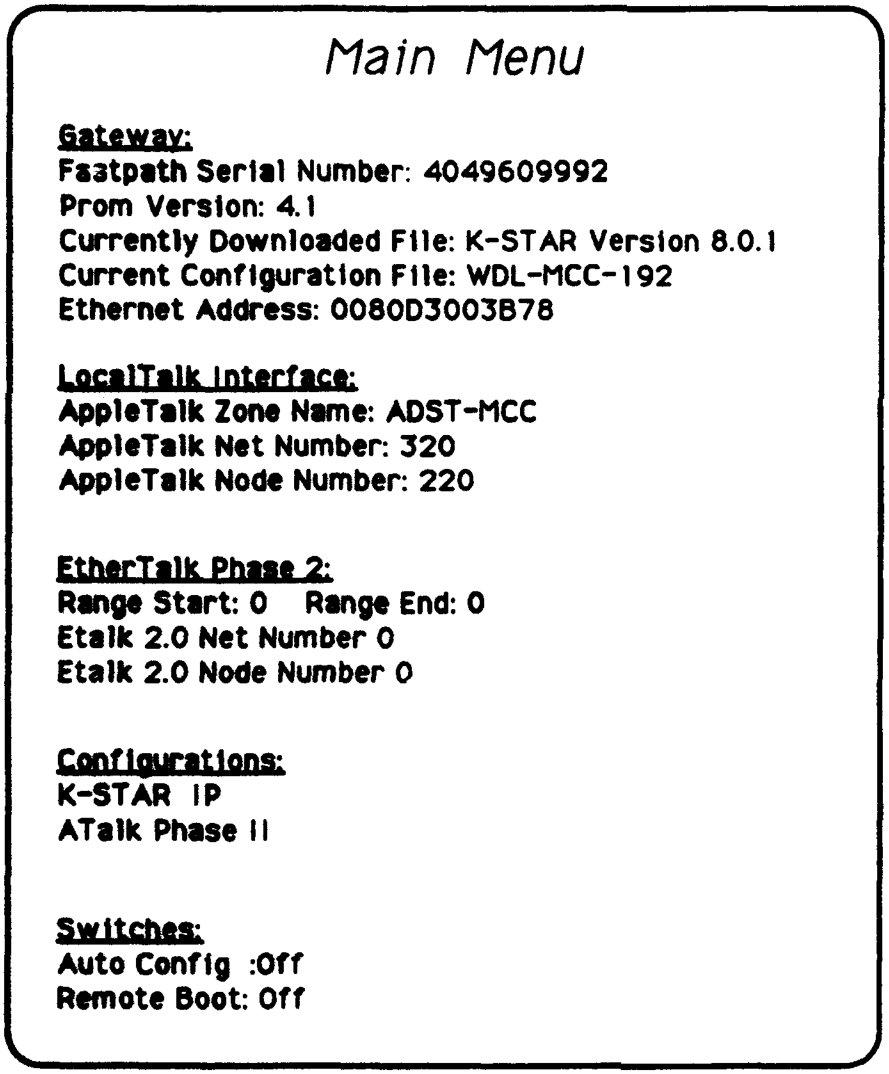
3) Check to see if appletalk zones are visible

```
> cd /usr3/cap60/samples
> getzones
```

2. SHIVA Setup

Shiva FastPath 4 (Gateway = [220])

IP - 192.67.225.10



```
Main Menu

Gateway:
Fastpath Serial Number: 4049609992
Prom Version: 4.1
Currently Downloaded File: K-STAR Version 8.0.1
Current Configuration File: WDL-MCC-192
Ethernet Address: 0080D3003B78

LocalTalk Interface:
AppleTalk Zone Name: ADST-MCC
AppleTalk Net Number: 320
AppleTalk Node Number: 220

EtherTalk Phase 2:
Range Start: 0   Range End: 0
Etalk 2.0 Net Number 0
Etalk 2.0 Node Number 0

Configurations:
K-STAR IP
ATalk Phase II

Switches:
Auto Config :Off
Remote Boot: Off
```

K - STAR IP

IP Information:
IP Address of FastPath Box - 192.67.225.10
IP Subnetwork Mask -255.255.255.0
IP Broadcast Address - 192.67.225.255
IP Address of Default Router - 0.0.0.0

Administrator:
IP Address of Administrator Host - 192.67.225.201

IPTalk Interface:
AppleTalk Zone Name : LORAL
AppleTalk Network Number: 1
Appletalk Node Number: 10

Number of IP Clients:
Dynamic: 0
Static: 0

Options Submenu (from Main)

Option Flags:
Off: all

Phase 1 Ethernet (not Phase 2II)
KIP - Kinetics IP

To Setup the Shiva:

1) Pause the MAC that you use to config. Shiva

2) Setup menu's as described above

3) Make sure "atis" is running on MIPs

4) Download K-Start 8.0.1

5) Type Go, that downloads these menu's to the Shiva

6) The log file should tell you that K-Star atalkab config. is complete